# xpertvr

### What scientific or technological uncertainties did you attempt to overcome? (Maximum 350 words)

XpertVR® exists in the virtual reality (VR) market space. We produce simulations for training purposes and tools for market research. We had the following technical uncertainties which required experimental development to overcome.

VR EMAIL DISTRIBUTION

We had to email reports to different types of people related to a VR experience. The VR experience was networked. The roles in the VR were participant and observer. The participant had to make decisions. The observer had to assess the decisions. The participant and the observer received two different email reports from the same networked VR experience in real-time. It was uncertain how to technologically tag, create, and send a message notification from inside the VR session, as no plug-ins exist in the VR framework. Unity, our development tool, had an existing plug-in that offered email handling, but it did not fit our use case. Amazon Web Services® also offer a generic email handler, but this needed severe modifications and additional code to work with our simulations. From our perspective, emails could be sent, but it was figuring out which data in scene needed to be sent back, and how that would be displayed meaningfully to the end user.

EYE TRACKING DATA COLLECTION IN VR

We had to develop data gathering methodologies and processes to access marketing data in created VRs. There is no firm framework or industry standard for this work, no working libraries or tool sets, and no data schema. As VR goes into the economy, its role becomes more functional, whereas, most of the work in VR has been recreational — gaming as an example. The market, and its academic corollaries like business and marketing schools, however, are looking for productivity from VR. Market surveys and consumer behavior studies are a natural match for VR. The field is still undeveloped, and, as such, much of the work is pioneering, and, therefore, uses experimental development as its driving force. We had to develop an eye-tracking tool for our VR scenarios to yield viable marketing data to our client. The previous eye tracking tool, Tobii, has been discontinued, leaving a gap in the market for eye-tracking VR software. Our team used an existing shader that used colour to display how long a cursor is hovering over a specific spot. This shader was modified to work with our VR eye-tracking scripts and placed on objects within our VR scene to develop a useful tool for researchers. This allowed individual objects to generate their own heat maps based on a user's time gazing at a specific object. As Tobii has built something similar in the past we knew it was possible to generate heat maps through VR eye tracking, and we took it upon ourselves to develop our own eye tracking tool in-house.

SIMULATION PORTABILITY

There are two domains to VR: simulation creation and simulation distribution. Creation is the domain of VR scene/simulation creators, software. Distribution is the domain of hardware. We employed a technique used in the gaming world — texture atlasing. It required extensive experimental development to fit this gaming technique. The results have been worthwhile, though, because we have created a portability to our scenarios by making them less memory intensive. By atlasing textures we were able to reduce draw calls in our simulations by, on average, 25-42 percent, which increases the frames per second (fps) for lower-end processors in headsets. Specifically, by way of example, we reduced 571 draw calls from a total of 1375 draw calls as part of a project, and therefore were able to achieve 60 fps on the atand-alone headset Oculus Quest 2®, which was running at only 40 fps when using the 1375 draw calls. Many of the new VR headsets are separating themselves from the personal computer (PC) and cell phones — which have been inserted into non-PC connected headsets — and therefore require lighter, or less memory and size intensive scenarios. Texture atlasing has pushed our VR scenarios into the emerging independent headset market.

### What work did you perform in the tax year to overcome the scientific or technological uncertainties described in line 242?

NEW EMAIL GENERATION AND DISTRIBUTION FOR NETWORKED VR

XpertVR designed seven crime scenes to be experienced with VR. They are accessed online with a headset at home. When the participant steps into the crime scene they are given the details on how to navigate around, interact with it, and the basic case file as understood is imparted to the student. This was designed for the Conestoga College Police Foundations diploma course. They are then free to do their classwork in the 3D VR crime scene. This is standard VR work, but what makes this work stand out is the email distribution system brought to a networked VR experience. The student is faced with decision-making situations throughout the VR. Those decisions are aggregated into a report for the student and a report for the professor. Although this might sound straightforward, in the VR world it is new territory.

The data collection was different for both parties. Each of the seven crime scenes had multiple sub-scenarios for experience and random user experience — no repetition — so each student would have a different and random experience with the VR. Each student has their VR experience assembled as a file which is sent to them, as well as a separate file sent to the professor. The file represents their decisions in the VR, and from that the student writes the final report, on which they are graded.

The unique aspect of this was the email component, specifically the separate reports for the two grades of participants, the student and the professor. There were no existing libraries for this kind of development. We diligently investigated and found nothing that suited our purpose. We tried several third-party solutions, but they did not meet our needs. Unity has a Plug in for sending information via email, but this did not suit our needs as the password/email used to create the email needed to be logged in on the receiving device in order to send the email. AWS offered another potential solution through a generic email sending tool, but in order for this to work for our use case we had to dynamically create HTML code based on the individual scenario a user went through to supply them with the right information upon completion. We had to build this from scratch. It represents a new email interface for networked VR applications. We understood the email could be sent, but our team had to design a way to send the email with meaningful information based on the VR users experience. This same email handler system will be easily implemented into future projects and grants us a new competitive advantage over our competitors.

EYE-TRACKING TOOL

The business faculty at Brock University is using XpertVR templates to conduct market research studies. XpertVR has made VR templates of retail environments as a way to reduce costs and increase reusability of its technological creations for our clients who focus on consumer behaviour testing. These clients will be able to leverage multiple retail environments quickly and effectively to test buying behaviours in these different retail environments. To date, VR companies have focused on custom one-time solutions for their clients, so we aim to create easy-to-use templates that drastically reduce the overall cost of launching a VR research study. One example is testing impulse buying. Placing products at the front of the VR store which people might only buy upon seeing. The research was designed to determine what forms buying patterns outside of what a user might need. As this is being built out, data collection tools are being created. These data collection models will make us faster and better for other clients and are completely reusable.

We built an eye-tracking tool, so as a VR user is going through a scenario a heat map is drawn when they stare at products. This is not new to the VR creation space, but ours works in tandem with a survey tool, the other most important thing for the marketing sector. The impetus behind productive VR is growing. Marketers want to know not only how they can access data from clients, but how they can do it without even asking questions, hence the eye-tracking tool. Only VR could deliver this, a new level of honesty is marketing surveys because the VR scenario is not asking your opinion, it's studying your attention through your eyes.

None of this work existed. There were no libraries or APIs to gather data from the tracked eye behavior of a VR user. We had to marry a data gathering based on eye-tracking, data categorizing for storage schema, and report preparation based on stored data. Each object within our simulations is tagged with information that will identify it in our data collection process. This is stored in a secure location to protect user data. We developed a system to export the data right to our AWS servers upon completion. This data can then be compiled into easily digestible dashboards for our researchers, or provide read-write data for their own analysis. This has significantly contributed to the practice of data gathering in VR, and places XpertVR in a leading position in this market space.

ATLASING TEXTURES

We experimented with texture atlasing, a common practice in the gaming world, to reduce the size and resource consumption of our scenarios. The repetition of small textures can be stored in a texture atlas. The texture atlas is treated as a single unit by the graphics processor, thus reducing resource utility. The VR world has made little use of this gaming tool. The APIs related to it are alien to the VR world. We had to leverage from the GPU APIs to marry this technique to VR. Normal 3D modeling pipelines require a modeler to create a material and texture for each model. These are then compiled into the VR scene with the model to provide the visual look. Once we began working with large amounts of models within a scene, our performance slowly declined. This began our research into how we could further optimize our simulations. After looking further into the game development industry, we discovered the method of atlasing textures, which we applied to our VR simulations. Our technical artist took all the materials from the models within a scene and merged similar materials to reduce the number of total materials in the scene. Because each material creates a new draw call that reduces the performance, by merging the materials we are able to drastically reduce the number of draw calls. A PC VR headset is plugged into computer, allowing for a high end, resource intensive experience. We leveraged from the gaming world to port our PC VR scenarios to standalone VR headsets. We achieved this by atlasing our textures. It has given us a methodology to single source our scenarios, which reduces memory use and CPU usage by approximately 25-42 percent.

### What scientific or technological advancements did you achieve or attempt to achieve as a result of the work described in line 244? (Maximum 350 words)

We advanced data gathering, sorting, and distribution, and the utility of these things in multiple VR deployments, as in the use cases of policing and marketing. We produced VR email distribution that emails reports to different types of people related to a networked VR experience. This significantly advanced, for us and for the industry, the utility data gathering and distribution based on VR experiences. We experimentally developed data gathering methodologies and processes to access and distribute marketing data in created VRs. This has multiple uses in multiple industries. It seriously advances our marketability, reputation, and gives us a competitive advantage in this market space. We experimented outside of the VR development space, importing techniques from the gaming world as associated with GPU APIs, and experimenting with those techniques until they worked in the VR space. This has allowed us to scale back the resource requirements for the emerging market of standalone headsets, advancing the principle of single sourcing, which is writing once and deploying to multiple platforms that have different resource economies. This, like the other experimental work described here, has given us competitive advantage.